

Reactive Forms

- Code and the logic resides in the component class
- No two way binding
- Well suited for complex scenarios
- Dynamic form fields.

Importing reactive form module

```
import { FormsModule, ReactiveFormsModuleModule } from  
        '@angular/forms';
```

```
imports: [
```

```
  ReactiveFormsModuleModule
```

```
],
```

Creating an reactive form

```
import { FormGroup, FormControl } from '@angular/forms';  
import { Component, OnInit } from '@angular/core';  
  
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})
```

```
export class AppComponent implements OnInit {  
  form = new FormGroup({  
    name: new FormControl(),  
    email: new FormControl(),  
    password: new FormControl()  
})
```

```
constructor() {}
```

```
ngOnInit() {}
```

```
  signup()
```

```
  {  
    console.log(this.form)  
  }
```

```
app.component.html  
<div class="container">  
  <form [formGroup]="form">  
    <div class="form-group">  
      <label>Name </label>  
      <input type="text" class="form-control"  
            formControlName="name">  
    </div>  
    <div class="form-group">  
      <label>Email </label>  
      <input type="text" class="form-control"  
            formControlName="email">  
    </div>  
    <div class="form-group">  
      <label>Password </label>  
      <input type="text" class="form-control"  
            formControlName="password">  
    </div>  
    <button (click)="signUp()">Submit</button>  
  </form>  
</div>
```

Output:-

Name

Email

Password

Submit

console screen

▼ Controls:

- email: FormControl
 - name: FormControl
 - password: FormControl
- proto--: Object

Template Driven Form: (minlength & maxlength)

app.component.html

```
<div class="container-fluid">
<form (ngSubmit)="onSubmit(form)" #form="ngForm">
  <div class="form-group">
    <label> Name </label>
    <input type="text" class="form-control"
      ngModel name="name" required minlength="5"
      maxlength="17" #name="ngMode">
  </div>
  <div class="alert alert-primary" *ngIf="name.touched
    && name.invalid">
    <div *ngIf="name.errors.required">
      <small> This field is required </small>
    </div>
    <div *ngIf="name.errors.minLength">
      <small> Min length is 5 </small>
    </div>
    <div *ngIf="name.errors.maxLength">
      <small> Max length is 7 </small>
    </div>
    <div *ngIf="name.errors.pattern">
      <small> This pattern is required </small>
    </div>
  </div>
  <button type="submit" class="btn btn-primary">
    Register </button>
  </form>
</div>
```

app.component.ts

```
import { Component }  
import { FormGroup }  
  
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})
```

```
export class AppComponent {  
  title = 'reactive Form'  
  onsubmit(form) {  
    console.log(form)  
  }  
}
```

Output:-

Name

AKS

minlength is 5.

Register

Console Screen

```
▼ Controls: Object  
  ▼ name: FormControl  
  ▼ errors:  
    ▼ minlength:  
      actualLength: 3  
      requiredLength: 5  
    ▶ -proto-: Object  
    ▶ -proto-: Object
```